In the Claims

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A method of generating an image using raster image processing, said image being generated based on a specification specifying said image, said specification containing data representing a plurality of objects in a page, said method comprising:

implementing a plurality of approaches, with each approach being designed to generate said image;

receiving said specification;

examining said specification to determine a number of resources required to generate said image of a page in each of said plurality of approaches;

selecting one of said plurality of approaches which requires an optimal number of resources;

generating said image of said page from said specification using said selected one of said plurality of approaches.

2 to 4. (Canceled)

5. (Currently Amended) The method of claim ~~4~~ 1, wherein:

said plurality of approaches includes rendering and screening via back-end screening and rendering and screening via pipelined screening; and

said examining comprises determining a minimum duration of time consumed by either said back-end screening or said pipelined screening.

6. (Original) The method of claim 5, wherein a system implementing said method contains a random access memory (RAM) and a cache, wherein said cache enables faster access to data to a processing unit, said determining a minimum duration further comprises determining an additional time required by said back-end screening approach due to storing rendered data in said RAM.

7. (Original) The method of claim 5, wherein said determining a minimum duration further comprises determining an additional time required by said pipelined screening due to the overlap of objects contained in said specification for said image.

8. (Original) The method of claim 5, wherein a system implementing said method contains a random access memory (RAM) and a cache, wherein said cache enables faster access to data to a processing unit, wherein said determining a minimum duration further comprises determining an additional time required by said pipelined screening when said processing unit accesses a code in cache enabling implementation of said pipelined screening, and an instruction cache miss results in accessing said code in said cache.

9. (Original) The method of claim 5, wherein a system implementing said method contains a random access memory (RAM) and a cache, wherein said cache enables faster access to data to a processing unit, said determining a minimum duration further comprises determining an additional time required by said pipelined screening if a data structure for a desired tile size does not fit in said cache, wherein said data structure is used in said pipelined screening.

10. (Canceled)

11. (Currently Amended) A system for generating an image using raster image processing, said image being generated based on a specification specifying said image, said specification containing data representing a plurality of objects in a page, said system comprising:

means for implementing a plurality of approaches, with each approach being designed to generate said image;

means for receiving said specification;

means for examining said specification to determine a number of resources required to generate said image of a page in each of said plurality of approaches;

means for selecting one of said plurality of approaches which requires an optimal number of resources;

means for generating said image of said page from said specification using said selected one of said plurality of approaches.

12 to 14. (Canceled)

15. (Currently Amended) The system of claim ~~14~~ 11, wherein:

said plurality of approaches includes rendering and screening via back-end screening and rendering and screening via pipelined screening; and

said means for examining comprises means for determining a minimum duration of time consumed by either said back-end screening or said pipelined screening.

16. (Canceled)

17. (Currently Amended) A computer program product for use with a computer system, said computer program product comprising a computer usable medium having computer readable program code means embodied in said medium generating an image using raster image processing, said image being generated based on a specification specifying said image, said specification containing data representing a plurality of objects of a page, said computer program product including:

computer readable program code means for implementing a plurality of approaches, with each approach being designed to generate said image;

computer readable program code means for receiving said specification;

computer readable program code means for examining said specification to determine a number of resources required to generate said image of a page in each of said plurality of approaches;

computer readable program code means for selecting one of said plurality of approaches which requires an optimal number of resources;

computer readable program code means for generating said image of said page from said specification using said selected one of said plurality of approaches.

18 to 20. (Canceled)

21. (Currently Amended) The computer program product of claim ~~20~~ 17, wherein:

said plurality of approaches includes rendering and screening via back-end screening and rendering and screening via pipelined screening; and

said computer readable program code means for examining comprises computer readable program code means for determining a minimum duration of time consumed by either said back-end screening or said pipelined screening.

22. (New) The method of claim 5, wherein:

said determining a minimum duration of time consumed by either said back-end screening or said pipelined screening calculates CLK as follows:

$$CLK = T_d - T_p - T_c$$

where T_d is the additional time required for back-end screening given by:

$$T_d = (\text{Pagesize} \times \text{DPSI}) \times 2 \,/\, (\text{Transfer-rate})$$

where: Pagesize is the area of the page; DPSI is the dots per square inch; and Transfer-rate is a memory data transfer rate;

where T_p is the difference in screening performance between pipeline screening and back-end screening given by:

$$T_p = (\sum A_i \times \text{DPSI} \times CLK_p) - (\text{Pagesize} \times \text{DPSI} \times CLK_b)$$

where: A_i is the bounding box area of i^{th} display list element i in square inches; CLK_p is the average number clock cycles required to process each pixel in pipeline screening; and CLK_b is

the average number clock cycles required to process each pixel in back-end screening;

where T_c is the difference in cache misses between pipeline screening back-end screening given by:

$$T_c = (\sum(I_n \times (C_{pi} - C_{bi}))) \times CLK_{cache}$$

where: In is the total number of element of type i in the display list; C_{pi} is the number of cache misses in one rendering for an element of type i in pipeline screening; C_{bi} is the number of cache misses in one rendering for element type i in back-end screening; and CLK_{cache} is the number of clock cycles required to serve one cache miss.

23. (New) The method of claim 5, wherein:

said determining a minimum duration of time consumed by either said back-end screening or said pipelined screening employing a predetermined screen tile height in screening calculates CLK as follows:

$$CLK = T_d - T_p - T_c - T_t$$

where T_d is the additional time required for back-end screening given by:

$$T_d = (\text{Pagesize} \times \text{DPSI}) \times 2 / (\text{Transfer-rate})$$

where: Pagesize is the area of the page; DPSI is the dots per square inch; and Transfer-rate is a memory data transfer rate;

where T_p is the difference in screening performance between pipeline screening and back-end screening given by:

$$T_p = (\sum A_i \times DPSI \times CLK_p) - (Pagesize \times DPSI \times CLK_b)$$

where: A_i is the bounding box area of i^{th} display list element i in square inches; CLK_p is the average number clock cycles required to process each pixel in pipeline screening; and CLK_b is the average number clock cycles required to process each pixel in back-end screening;

where T_c is the difference in cache misses between pipeline screening back-end screening given by:

$$T_c = (\sum(_{In} \times (C_{pi} - C_{bi}))) \times CLK_{cache}$$

where: In is the total number of element of type i in the display list; C_{pi} is the number of cache misses in one rendering for an element of type i in pipeline screening; C_{bi} is the number of cache misses in one rendering for element type i in back-end screening; and CLK_{cache} is the number of clock cycles required to serve one cache miss; and

where: T_t is the number of excess clock cycles between pipelined screening and back-end screening due to a screening table not being able to fit in a cache when the object height is less than the predetermined screen tile height given by:

$$T_t = (\sum a_i \times DPSI) \times CLK_{tp})$$

where: a_i is bounding box area of display list element i; and CLK_{tp} is the number of clock cycles to process a cache miss.

24. (New) The method of claim 5, wherein:

said determining a minimum duration of time consumed by either said back-end screening or said pipelined screening employing a predetermined screen tile height in screening calculates CLK as follows:

$$CLK = T_d - T_p - T_c - T_t$$

where T_d is the additional time required for back-end screening given by:

$$T_d = (\text{Pagesize} \times \text{DPSI}) \times 2 \; / \; (\text{Transfer-rate})$$

where: Pagesize is the area of the page; DPSI is the dots per square inch; and Transfer-rate is a memory data transfer rate;

where T_p is the difference in screening performance between pipeline screening and back-end screening given by:

$$T_p = (\sum A_i \times \text{DPSI} \times CLK_p) - (\text{Pagesize} \times \text{DPSI} \times CLK_b)$$

where: A_i is the bounding box area of i^{th} display list element i in square inches; CLK_p is the average number clock cycles required to process each pixel in pipeline screening; and CLK_b is the average number clock cycles required to process each pixel in back-end screening;

where T_c is the difference in cache misses between pipeline screening back-end screening given by:

$$T_c = (\sum (I_n \times (C_{pi} - C_{bi}))) \times CLK_{cache}$$

where: In is the total number of element of type i in the display list; C_{pi} is the number of cache misses in one rendering for an element of type i in pipeline screening; C_{bi} is the number of cache misses in one rendering for element type i in back-end screening; and CLK_{cache} is the number of clock cycles required to serve one cache miss; and

where: T_t is the number of excess clock cycles between pipelined screening and back-end screening due to a screening table not being able to fit in a cache when the object height is greater than the predetermined screen tile height given by:

$$T_t = (\sum h_i \times DPSI \times T_w) \times CLK_{tp}$$

where: h_i is the height of display list element i; T_w is the width of the screen; and CLK_{tp} is the number of clock cycles to process a cache miss.

25. (New) The system of claim 15, further comprising:

a random access memory (RAM); and

a cache, wherein said cache enables faster access to data to a processing unit; and

wherein said means for determining a minimum duration further comprises means for determining an additional time required by said back-end screening approach due to storing rendered data in said RAM.

26. (New) The method of claim 15, wherein:

said means for determining a minimum duration further comprises means for determining an additional time required by said pipelined screening due to the overlap of objects contained in said specification for said image.

27. (New) The method of claim 15, further comprising:

a random access memory (RAM);

a cache, wherein said cache enables faster access to data to a processing unit than said random access memory; and

wherein said means for determining a minimum duration further comprises means for determining an additional time required by said pipelined screening when said processing unit accesses a code in cache enabling implementation of said pipelined screening, and an instruction cache miss results in accessing said code in said cache.

28. (New) The method of claim 15, further comprising:

a random access memory (RAM);

a cache, wherein said cache enables faster access to data to a processing unit than said random access memory (RAM); and

wherein said means for determining a minimum duration further comprises means for determining an additional time required by said pipelined screening if a data structure for a desired tile size does not fit in said cache, wherein said data structure is used in said pipelined screening.

29. (New) The method of claim 15, wherein:

said determining a minimum duration of time consumed by either said back-end screening or said pipelined screening calculates CLK as follows:

$$CLK = T_d - T_p - T_c$$

where T_d is the additional time required for back-end screening given by:

$$T_d = (\text{Pagesize} \times \text{DPSI}) \times 2 / (\text{Transfer-rate})$$

where: Pagesize is the area of the page; DPSI is the dots per square inch; and Transfer-rate is a memory data transfer rate;

where T_p is the difference in screening performance between pipeline screening and back-end screening given by:

$$T_p = (\sum A_i \times \text{DPSI} \times CLK_p) - (\text{Pagesize} \times \text{DPSI} \times CLK_b)$$

where: A_i is the bounding box area of i^{th} display list element i in square inches; CLK_p is the average number clock cycles required to process each pixel in pipeline screening; and CLK_b is the average number clock cycles required to process each pixel in back-end screening;

where T_c is the difference in cache misses between pipeline screening back-end screening given by:

$$T_c = (\sum (I_n \times (C_{pi} - C_{bi}))) \times CLK_{cache}$$

where: In is the total number of element of type i in the display list; C_{pi} is the number of cache misses in one rendering for an element of type i in pipeline screening; C_{bi} is the number of cache misses in one rendering for element type i in back-end screening; and CLK_{cache} is the number of clock cycles required to serve one cache miss.

30. (New) The method of claim 15, wherein:

said determining a minimum duration of time consumed by either said back-end screening or said pipelined screening

employing a predetermined screen tile height in screening caclulates CLK as follows:

$$CLK = T_d - T_p - T_c - T_t$$

where T_d is the additional time required for back-end screening given by:

$$T_d = (\text{Pagesize} \times \text{DPSI}) \times 2 / (\text{Transfer-rate})$$

where: Pagesize is the area of the page; DPSI is the dots per square inch; and Transfer-rate is a memory data transfer rate;

where T_p is the difference in screening performance between pipeline screening and back-end screening given by:

$$T_p = (\sum A_i \times \text{DPSI} \times CLK_p) - (\text{Pagesize} \times \text{DPSI} \times CLK_b)$$

where: A_i is the bounding box area of i^{th} display list element i in square inches; CLK_p is the average number clock cycles required to process each pixel in pipeline screening; and CLK_b is the average number clock cycles required to process each pixel in back-end screening;

where T_c is the difference in cache misses between pipeline screening back-end screening given by:

$$T_c = (\sum (I_n \times (C_{pi} - C_{bi}))) \times CLK_{cache}$$

where: In is the total number of element of type i in the display list; C_{pi} is the number of cache misses in one rendering for an element of type i in pipeline screening; C_{bi} is the number of cache misses in one rendering for element type i in back-end

screening; and CLK_{cache} is the number of clock cycles required to serve one cache miss; and

where: T_t is the number of excess clock cycles between pipelined screening and back-end screening due to a screening table not being able to fit in a cache when the object height is less than the predetermined screen tile height given by:

$$T_t = (\sum a_i \times DPSI) \times CLK_{tp}$$

where: a_i is bounding box area of display list element i; and CLK_{tp} is the number of clock cycles to process a cache miss.

31. (New) The method of claim 15, wherein:

said determining a minimum duration of time consumed by either said back-end screening or said pipelined screening employing a predetermined screen tile height in screening calculates CLK as follows:

$$CLK = T_d - T_p - T_c - T_t$$

where T_d is the additional time required for back-end screening given by:

$$T_d = (Pagesize \times DPSI) \times 2 \; / \; (Transfer\text{-}rate)$$

where: Pagesize is the area of the page; DPSI is the dots per square inch; and Transfer-rate is a memory data transfer rate;

where T_p is the difference in screening performance between pipeline screening and back-end screening given by:

$$T_p = (\sum A_i \times DPSI \times CLK_p) - (Pagesize \times DPSI \times CLK_b)$$

where: A_i is the bounding box area of i^{th} display list element i in square inches; CLK_p is the average number clock cycles required to process each pixel in pipeline screening; and CLK_b is the average number clock cycles required to process each pixel in back-end screening;

where T_c is the difference in cache misses between pipeline screening back-end screening given by:

$$T_c = (\sum(_{In} \times (C_{pi} - C_{bi}))) \times CLK_{cache}$$

where: In is the total number of element of type i in the display list; C_{pi} is the number of cache misses in one rendering for an element of type i in pipeline screening; C_{bi} is the number of cache misses in one rendering for element type i in back-end screening; and CLK_{cache} is the number of clock cycles required to serve one cache miss; and

where: T_t is the number of excess clock cycles between pipelined screening and back-end screening due to a screening table not being able to fit in a cache when the object height is greater than the predetermined screen tile height given by:

$$T_t = (\sum h_i \times DPSI \times T_w) \times CLK_{tp}$$

where: h_i is the height of display list element i; T_w is the width of the screen; and CLK_{tp} is the number of clock cycles to process a cache miss.

32. (New) A method of generating an image using raster image processing, said image being generated based on a specification specifying said image, said specification

containing data representing a plurality of objects in a page, said method comprising:

implementing a plurality of approaches, with each approach being designed to generate said image;

receiving said specification;

examining said specification to determine a number of resources required to generate said image of a subset of a page including plural objects in each of said plurality of approaches;

selecting one of said plurality of approaches which requires an optimal number of resources;

generating said image of said page from said specification using said selected one of said plurality of approaches.

33. (New) The method of claim 32, wherein:

said subset of a page including plural objects consists of a band of a predetermined number of adjacent horizontal lines.

34. (New) The method of claim 32, wherein:

said subset of a page including plural objects consists of a subband of a rectangle of a predetermined number of adjacent pixels in a predetermined number of adjacent horizontal lines.

35. (New) The method of claim 32, wherein:

said plurality of approaches includes rendering and screening via back-end screening and rendering and screening via pipelined screening;

said examining comprises determining a minimum duration of time consumed by either said back-end screening or said pipelined screening; and

wherein a system implementing said method contains a random access memory (RAM) and a cache, wherein said cache enables

faster access to data to a processing unit, said determining a minimum duration further comprises determining an additional time required by said back-end screening approach due to storing rendered data in said RAM.

36. (New) The method of claim 32, wherein:

said plurality of approaches includes rendering and screening via back-end screening and rendering and screening via pipelined screening;

said examining comprises determining a minimum duration of time consumed by either said back-end screening or said pipelined screening; and

wherein said determining a minimum duration further comprises determining an additional time required by said pipelined screening due to the overlap of objects contained in said specification for said image.

37. (New) The method of claim 32, wherein:

said plurality of approaches includes rendering and screening via back-end screening and rendering and screening via pipelined screening;

said examining comprises determining a minimum duration of time consumed by either said back-end screening or said pipelined screening; and

wherein a system implementing said method contains a random access memory (RAM) and a cache, wherein said cache enables faster access to data to a processing unit, wherein said determining a minimum duration further comprises determining an additional time required by said pipelined screening when said processing unit accesses a code in cache enabling implementation

of said pipelined screening, and an instruction cache miss results in accessing said code in said cache.

38. (New) The method of claim 32, wherein:

said plurality of approaches includes rendering and screening via back-end screening and rendering and screening via pipelined screening;

said examining comprises determining a minimum duration of time consumed by either said back-end screening or said pipelined screening; and

wherein a system implementing said method contains a random access memory (RAM) and a cache, wherein said cache enables faster access to data to a processing unit, said determining a minimum duration further comprises determining an additional time required by said pipelined screening if a data structure for a desired tile size does not fit in said cache, wherein said data structure is used in said pipelined screening.

39. (New) The method of claim 32, wherein:

said determining a minimum duration of time consumed by either said back-end screening or said pipelined screening calculates CLK as follows:

$$CLK = T_d - T_p - T_c$$

where T_d is the additional time required for back-end screening given by:

$$T_d = (\text{Pagesize} \times \text{DPSI}) \times 2 / (\text{Transfer-rate})$$

where: Pagesize is the area of the page; DPSI is the dots per square inch; and Transfer-rate is a memory data transfer rate;

where T_p is the difference in screening performance between pipeline screening and back-end screening given by:

$$T_p = (\sum A_i \times DPSI \times CLK_p) - (Pagesize \times DPSI \times CLK_b)$$

where: A_i is the bounding box area of i^{th} display list element i in square inches; CLK_p is the average number clock cycles required to process each pixel in pipeline screening; and CLK_b is the average number clock cycles required to process each pixel in back-end screening;

where T_c is the difference in cache misses between pipeline screening back-end screening given by:

$$T_c = (\sum (I_n \times (C_{pi} - C_{bi}))) \times CLK_{cache}$$

where: In is the total number of element of type i in the display list; C_{pi} is the number of cache misses in one rendering for an element of type i in pipeline screening; C_{bi} is the number of cache misses in one rendering for element type i in back-end screening; and CLK_{cache} is the number of clock cycles required to serve one cache miss.

40. (New) The method of claim 32, wherein:

said determining a minimum duration of time consumed by either said back-end screening or said pipelined screening employing a predetermined screen tile height in screening calculates CLK as follows:

$$CLK = T_d - T_p - T_c - T_t$$

where T_d is the additional time required for back-end screening given by:

$$T_d = (\text{Pagesize} \times \text{DPSI}) \times 2 \; / \; (\text{Transfer-rate})$$

where: Pagesize is the area of the page; DPSI is the dots per square inch; and Transfer-rate is a memory data transfer rate;

where T_p is the difference in screening performance between pipeline screening and back-end screening given by:

$$T_p = (\sum A_i \times \text{DPSI} \times CLK_p) - (\text{Pagesize} \times \text{DPSI} \times CLK_b)$$

where: A_i is the bounding box area of i^{th} display list element i in square inches; CLK_p is the average number clock cycles required to process each pixel in pipeline screening; and CLK_b is the average number clock cycles required to process each pixel in back-end screening;

where T_c is the difference in cache misses between pipeline screening back-end screening given by:

$$T_c = (\sum (In \times (C_{pi} - C_{bi}))) \times CLK_{cache}$$

where: In is the total number of element of type i in the display list; C_{pi} is the number of cache misses in one rendering for an element of type i in pipeline screening; C_{bi} is the number of cache misses in one rendering for element type i in back-end screening; and CLK_{cache} is the number of clock cycles required to serve one cache miss; and

where: T_t is the number of excess clock cycles between pipelined screening and back-end screening due to a screening

table not being able to fit in a cache when the object height is less than the predetermined screen tile height given by:

$$T_t = (\sum a_i \times DPSI) \times CLK_{tp}$$

where: a_i is bounding box area of display list element i; and CLK_{tp} is the number of clock cycles to process a cache miss.

41. (New) The method of claim 32, wherein:

said determining a minimum duration of time consumed by either said back-end screening or said pipelined screening employing a predetermined screen tile height in screening calculates CLK as follows:

$$CLK = T_d - T_p - T_c - T_t$$

where T_d is the additional time required for back-end screening given by:

$$T_d = (Pagesize \times DPSI) \times 2 / (Transfer\text{-}rate)$$

where: Pagesize is the area of the page; DPSI is the dots per square inch; and Transfer-rate is a memory data transfer rate;

where T_p is the difference in screening performance between pipeline screening and back-end screening given by:

$$T_p = (\sum A_i \times DPSI \times CLK_p) - (Pagesize \times DPSI \times CLK_b)$$

where: A_i is the bounding box area of i^{th} display list element i in square inches; CLK_p is the average number clock cycles required to process each pixel in pipeline screening; and CLK_b is

the average number clock cycles required to process each pixel in back-end screening;

where T_c is the difference in cache misses between pipeline screening back-end screening given by:

$$T_c = (\sum(I_n \times (C_{pi} - C_{bi}))) \times CLK_{cache}$$

where: In is the total number of element of type i in the display list; C_{pi} is the number of cache misses in one rendering for an element of type i in pipeline screening; C_{bi} is the number of cache misses in one rendering for element type i in back-end screening; and CLK_{cache} is the number of clock cycles required to serve one cache miss; and

where: T_t is the number of excess clock cycles between pipelined screening and back-end screening due to a screening table not being able to fit in a cache when the object height is greater than the predetermined screen tile height given by:

$$T_t = (\sum h_i \times DPSI \times T_w) \times CLK_{tp}$$

where: h_i is the height of display list element i; T_w is the width of the screen; and CLK_{tp} is the number of clock cycles to process a cache miss.

42. (New) A system for generating an image using raster image processing, said image being generated based on a specification specifying said image, said specification containing data representing a plurality of objects in a page, said system comprising:

means for implementing a plurality of approaches, with each approach being designed to generate said image;

means for receiving said specification;

means for examining said specification to determine a number of resources required to generate said image of a subset of a page including plural objects in each of said plurality of approaches;

means for selecting one of said plurality of approaches which requires an optimal number of resources;

means for generating said image of said page from said specification using said selected one of said plurality of approaches.

43. (New) The method of claim 42, wherein:

said subset of a page including plural objects consists of a band of a predetermined number of adjacent horizontal lines.

44. (New) The method of claim 42, wherein:

said subset of a page including plural objects consists of a subband of a rectangle of a predetermined number of adjacent pixels in a predetermined number of adjacent horizontal lines.

45. (New) The system of claim 42, further comprising:

a random access memory (RAM);

a cache, wherein said cache enables faster access to data to a processing unit;

wherein said plurality of approaches includes rendering and screening via back-end screening and rendering and screening via pipelined screening;

wherein said means for examining comprises means for determining a minimum duration of time consumed by either said back-end screening or said pipelined screening; and

wherein said means for determining a minimum duration further comprises means for determining an additional time required by said back-end screening approach due to storing rendered data in said RAM.

46. (New) The method of claim 42, wherein:

said means for determining a minimum duration further comprises means for determining an additional time required by said pipelined screening due to the overlap of objects contained in said specification for said image.

47. (New) The method of claim 42, further comprising:

a random access memory (RAM);

a cache, wherein said cache enables faster access to data to a processing unit than said random access memory; and

wherein said means for determining a minimum duration further comprises means for determining an additional time required by said pipelined screening when said processing unit accesses a code in cache enabling implementation of said pipelined screening, and an instruction cache miss results in accessing said code in said cache.

48. (New) The method of claim 42, further comprising:

a random access memory (RAM);

a cache, wherein said cache enables faster access to data to a processing unit than said random access memory (RAM); and

wherein said means for determining a minimum duration further comprises means for determining an additional time required by said pipelined screening if a data structure for a desired tile size does not fit in said cache, wherein said data structure is used in said pipelined screening.

49. (New) The method of claim 42, wherein:

said determining a minimum duration of time consumed by either said back-end screening or said pipelined screening calculates CLK as follows:

$$CLK = T_d - T_p - T_c$$

where T_d is the additional time required for back-end screening given by:

$$T_d = (\text{Pagesize} \times \text{DPSI}) \times 2 \; / \; (\text{Transfer-rate})$$

where: Pagesize is the area of the page; DPSI is the dots per square inch; and Transfer-rate is a memory data transfer rate;

where T_p is the difference in screening performance between pipeline screening and back-end screening given by:

$$T_p = (\sum A_i \times \text{DPSI} \times CLK_p) - (\text{Pagesize} \times \text{DPSI} \times CLK_b)$$

where: A_i is the bounding box area of i^{th} display list element i in square inches; CLK_p is the average number clock cycles required to process each pixel in pipeline screening; and CLK_b is the average number clock cycles required to process each pixel in back-end screening;

where T_c is the difference in cache misses between pipeline screening back-end screening given by:

$$T_c = (\sum (r_n \times (C_{pi} - C_{bi}))) \times CLK_{cache}$$

where: In is the total number of element of type i in the display list; C_{pi} is the number of cache misses in one rendering for an element of type i in pipeline screening; C_{bi} is the number of cache misses in one rendering for element type i in back-end screening; and CLK_{cache} is the number of clock cycles required to serve one cache miss.

50. (New) The method of claim 42, wherein:

said determining a minimum duration of time consumed by either said back-end screening or said pipelined screening employing a predetermined screen tile height in screening caclulates CLK as follows:

$$CLK = T_d - T_p - T_c - T_t$$

where T_d is the additional time required for back-end screening given by:

$$T_d = (\text{Pagesize} \times \text{DPSI}) \times 2 \;/\; (\text{Transfer-rate})$$

where: Pagesize is the area of the page; DPSI is the dots per square inch; and Transfer-rate is a memory data transfer rate;

where T_p is the difference in screening performance between pipeline screening and back-end screening given by:

$$T_p = (\sum A_i \times \text{DPSI} \times CLK_p) - (\text{Pagesize} \times \text{DPSI} \times CLK_b)$$

where: A_i is the bounding box area of i^{th} display list element i in square inches; CLK_p is the average number clock cycles required to process each pixel in pipeline screening; and CLK_b is

the average number clock cycles required to process each pixel in back-end screening;

where T_c is the difference in cache misses between pipeline screening back-end screening given by:

$$T_c = (\sum(I_n \times (C_{pi} - C_{bi}))) \times CLK_{cache}$$

where: In is the total number of element of type i in the display list; C_{pi} is the number of cache misses in one rendering for an element of type i in pipeline screening; C_{bi} is the number of cache misses in one rendering for element type i in back-end screening; and CLK_{cache} is the number of clock cycles required to serve one cache miss; and

where: T_t is the number of excess clock cycles between pipelined screening and back-end screening due to a screening table not being able to fit in a cache when the object height is less than the predetermined screen tile height given by:

$$T_t = (\sum a_i \times DPSI) \times CLK_{tp})$$

where: a_i is bounding box area of display list element i; and CLK_{tp} is the number of clock cycles to process a cache miss.

51. (New) The method of claim 42, wherein:

said determining a minimum duration of time consumed by either said back-end screening or said pipelined screening employing a predetermined screen tile height in screening calculates CLK as follows:

$$CLK = T_d - T_p - T_c - T_t$$

where T_d is the additional time required for back-end screening given by:

$$T_d = (\text{Pagesize} \times \text{DPSI}) \times 2 \; / \; (\text{Transfer-rate})$$

where: Pagesize is the area of the page; DPSI is the dots per square inch; and Transfer-rate is a memory data transfer rate;

where T_p is the difference in screening performance between pipeline screening and back-end screening given by:

$$T_p = (\sum A_i \times \text{DPSI} \times CLK_p) - (\text{Pagesize} \times \text{DPSI} \times CLK_b)$$

where: A_i is the bounding box area of i^{th} display list element i in square inches; CLK_p is the average number clock cycles required to process each pixel in pipeline screening; and CLK_b is the average number clock cycles required to process each pixel in back-end screening;

where T_c is the difference in cache misses between pipeline screening back-end screening given by:

$$T_c = (\sum (I_n \times (C_{pi} - C_{bi}))) \times CLK_{cache}$$

where: In is the total number of element of type i in the display list; C_{pi} is the number of cache misses in one rendering for an element of type i in pipeline screening; C_{bi} is the number of cache misses in one rendering for element type i in back-end screening; and CLK_{cache} is the number of clock cycles required to serve one cache miss; and

where: T_t is the number of excess clock cycles between pipelined screening and back-end screening due to a screening

table not being able to fit in a cache when the object height is greater than the predetermined screen tile height given by:

$$T_t = (\sum h_i \times DPSI \times T_w) \times CLK_{tp}$$

where: h_i is the height of display list element i; T_w is the width of the screen; and CLK_{tp} is the number of clock cycles to process a cache miss.